


Working with Dynamic HTML (DHTML)

Objectives

- ▶ Define Dynamic HTML
- ▶ Understand the building blocks of DHTML
- ▶ Tour DHTML pages
- ▶ Understand the DHTML Object Model
- ▶ Understand browser variability
- ▶ Design DHTML pages
- ▶ Research code architecture
- ▶ Keep up with DHTML changes

Once you have an understanding of standard HTML and Web page design, you can create well-structured Web pages that use effective style combinations and that allow basic user input. However, recent innovations in Web page design and scripting, collectively known as **Dynamic HTML** (or **DHTML** for short), have revolutionized Web page design. DHTML has greatly increased the degree of interactivity possible in Web page design. With DHTML, your Web pages are enlivened as text and graphics change color, grow, shrink, and move on and off the page in response to user actions.  Lydia Burgos, who works in the information systems department at Nomad Ltd, has read about Dynamic HTML and wants to explore using it in her company's Web pages. She starts with some research to learn about what DHTML is and how it works.





Defining DHTML

During the early 1990's, all Web pages were simple documents that users downloaded and viewed on their local computers. Each Web page's interactivity was limited to hyperlinks, which opened other Web pages, opened new mail messages, or ran scripts on the server. Web pages that fit this description are known as **static HTML**. Today, however, many Web pages respond to and even interact with the user by changing their appearances based on user actions. Such pages use **dynamic HTML**, which describes a varied set of technologies that allow almost-immediate response to user actions in a Web page without accessing the Internet server. In her research, Lydia learns of several broad categories of design that DHTML allows.

Details



Dynamic style

When you create a page using standard HTML coding, you specify a style for each text element. These styles remain the same, regardless of user actions. The one exception to this is hyperlinks; their color may be changed by the browser if you have followed them recently. However, when you create a page using DHTML, you can incorporate styles—including font size, typeface, and color—that change immediately in response to user actions, such as moving the mouse pointer over a heading. This feature, known as **dynamic style**, allows your pages to emphasize an area when a user shows interest in it, without flooding the page with distracting large font sizes or bright colors. Figure I-1 provides an example of dynamic style. Notice on this DHTML page that the text color has changed, which is the result of DHTML. If the user selects this hyperlink, the color will change again to show that it has been viewed already.



Dynamic content

A DHTML Web page can display different content based on a user's activities, which is a feature known as **dynamic content**. Instead of taking the time to request, download, and display a new Web page (as standard HTML coding would do), DHTML utilities can simply hide or display blocks of text or other elements in the current page. This aspect of DHTML allows you to create a simple, well-organized, and visually appealing page that can instantly display extra information when the user is likely to be interested in it. Figure I-1 provides an example of dynamic content. Notice the message displayed in the status window, which is the result of DHTML.



Data-awareness

Standard HTML tools allow your Web pages to download chunks of information, such as database contents, from a Web server as a user requests access to them. With DHTML, this process is instantaneous for the user; for example, a DHTML Web page could be designed to download a complete database but then display only the information the user wants to view. A Web page equipped to work with data in this way is termed **data aware**, which means the user can work with information from a Web server without adding to Internet traffic by repeatedly requesting additional pieces of information. Also, data awareness can allow the user to manipulate and change the information right in the browser window.



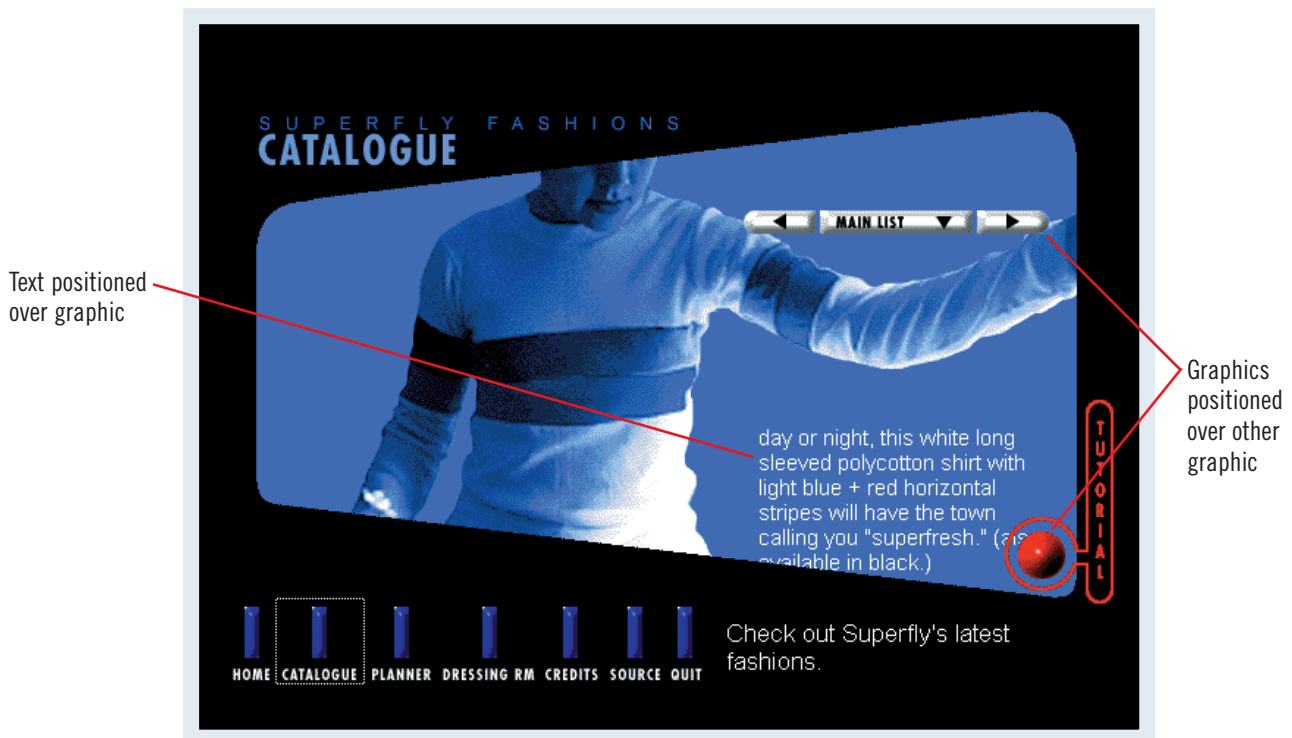
Positioning

As with other formatting options, static HTML leaves many of the choices regarding the positioning of elements in a Web page to the browser's discretion. In addition to causing pages to display nonuniformly and unpredictably on different browsers, this aspect of HTML has prevented Web page design from rivaling the intricacy inherent in the best layouts of other media, such as magazines. DHTML represents an important step toward changing this by allowing Web page designers to specify precisely the location of all page elements, a feature known as **positioning**, which is unavailable in standard HTML. The Web page in Figure I-2, which is from an online tutorial on DHTML, uses DHTML to position text in combinations not possible with static HTML.

FIGURE I-1: Dynamic Web page



FIGURE I-2: Web page formatted with DHTML positioning



Understanding the Building Blocks of DHTML

The creation of simple Web pages, while drawing on organization, design, and content-production skills, uses only HTML for arrangement and display in a browser. By contrast, DHTML is not a language or even a single technology, but, rather, a collection of Web page tools that, when used in various combinations, let designers create the effects specified in the previous lesson. As she reads more about DHTML, Lydia learns that DHTML is comprised of two main tools that work in tandem with standard HTML. These DHTML tools are included in the Web page source code shown in Figure I-3.

Details



Client-side scripts

Scripts are small programs that can be triggered by a user's action on a Web page. In the early days of the Web, browsers allowed only the use of **server-side scripts**, or scripts that were stored and run on the Web server. Using these server-side scripts was similar to triggering a hyperlink. Each time a server-side script was run, the Web page sent a message to the server instructing it to run the script. Users had to wait as the browser downloaded the results of the server-side script. The lag time involved in this setup made features such as dynamic content and dynamic style impractical. Recent versions of browsers have allowed Web page designers to create **client-side scripts**, or scripts that the browser itself interprets and runs. Client-side scripts are a key element in allowing DHTML to respond immediately to user actions. For example, the client-side script at the bottom of Figure I-3 changes the text color and adds text to the status window when the mouse pointer moves over certain text.



Cascading Style Sheets (CSS)

In standard HTML, you assign styles and properties to elements of your page—text blocks, images, and other objects—through HTML tags. This system means that each element has its own set of properties. Even if two elements share the same properties, you must assign them separately and make any subsequent changes to each element. Although you can assign similar properties to groups of elements using defined styles, such as `<H1>`, these styles are defined on each user's browser, and thus you cannot predict exactly how a viewer will see your page in the browser window. **Cascading Style Sheets (CSS)** is a tool that allows you to specify attributes such as color and font size for all page elements marked by a specific tag, name, or ID. CSS not only gives designers a more efficient way to specify style but also more control over an object's attributes as well as how each object should be displayed in certain situations. For example, the Cascading Style Sheet in Figure I-3 assigns attributes to various tags. All text marked with these tags, such as `` and `<P>`, will display the attributes defined for this tag in the style sheet.

FIGURE I-3: Code for Web page incorporating DHTML tools

Cascading Style Sheet	<pre><STYLE TYPE="text/css"> body {background:navy; color:white} LI {list-style-image: none; list-style: none} UL.toc {display:none} UL.expanded {display:block} A.select {color:white; background:blue} .over {color:red} P {margin-top:0; margin-bottom:0} </STYLE></pre>
Client-side script	<pre><SCRIPT LANGUAGE="JavaScript"> <!-- var curSelection = null; function setStyle(src, toClass) { if (null != src) src.className = toClass; } function mouseEnters() { if ((curSelection != event.toElement) && ("A" == event.toElement.tagName)) { setStyle(event.toElement, "over"); window.status="Fall registration begins August 15, 2000." } }</pre>




Proprietary features

Both Netscape and Microsoft have each introduced unique features, known as **proprietary features**, into their browsers. For example, Netscape Navigator 4 allows use of the <LAYER> tag to overlap screen elements easily. Microsoft Internet Explorer 4 supports the embedding of external tables in a Web page, as well as a set of features that affect element appearance in complex ways. Eventually, some of these technologies

become part of new international Web page standards. When proprietary features become part of the industry standard, they eventually become supported by the major browsers and are then no longer considered proprietary features. However, proprietary features that are supported by only one of the two major browsers are most useful only in single-browser settings, such as intranets whose users all run the same browser.



Touring DHTML Pages

Although DHTML technology may sound intriguing, viewing and interacting with it is the only way to get a true sense of its impact and capabilities. Looking at existing pages, both successful and not, is also a useful way to begin planning the features you want to include in your own pages.  Lydia has downloaded several sample Web pages that incorporate features she has researched. She opens and tests them as she begins collecting ideas for updating the Nomad Ltd Web site.

Steps 1 2 3 4

Trouble?

If you are using Navigator, some text on the page may be arranged differently from that shown in the figures. However, all of the features of the page should still work.

1. Start your Web browser program, open the file **HTML I-1.htm**, then scroll down the page to view its layout

As Figure I-4 shows, this page contains several blocks of text positioned around the page; each of these is an example of DHTML positioning. The designer of this Web page created the sidebar along the right edge of the screen by using DHTML style specifications to position the text, specify its width, and specify a background color for the text block.

2. Scroll down the page until the heading **Blue Ray** appears in your document window, then move your mouse pointer over the heading **Blue Ray**

If you are using Internet Explorer 4, notice that the text color changes from black to purple and that the text size increases—an example of **dynamic style**. Netscape Navigator 4 does not support most dynamic styles and shows no change when your mouse pointer is over this heading.

3. Click the heading **Blue Ray**

A paragraph of detailed information appears beneath the heading, without the page reloading, as shown in Figure I-5. This is an example of **dynamic content** because user activity can affect the page content.

4. Watch the text in the status window

A message continuously scrolls across the status window. This feature, created by a script, is another example of dynamic content.

5. Scroll to the top of the page, and move your mouse pointer over one of the links under the heading “Learn more about Jim’s!”

As the pointer moves over link text or an image, the link image changes. When the pointer moves off the link, the image returns to its original appearance, which is an instance of dynamic content. Rather than simply changing a graphic’s display properties, the position of the pointer over the link triggers a script that changes the source of the image in the image tag. The pointer movement causes the image to toggle between two different source files.

QuickTip

Currently, data binding is a proprietary feature of Internet Explorer 4, but an extension is available from the Microsoft Web site that allows Netscape Navigator 4 to display this and other Internet Explorer 4 features.

6. If you are using Internet Explorer, open the file **HTML I-2.htm** and scroll to the bottom of the page

As Figure I-6 shows, this page contains a data table. Unlike standard HTML tables, however, this table was generated from an external file as the Web page opened. Linking a Web page to an external data file is known as **data binding**. If you added or changed records in the external file, they would be reflected in the Web page the next time you opened it without requiring any changes in the Web page’s code. A related feature, known as **data-awareness**, allows a Web page to load all the records from a database but display only some of them. Then, a user can access any record instantly without needing to download more information to the browser.

FIGURE I-4: Sample DHTML Web page

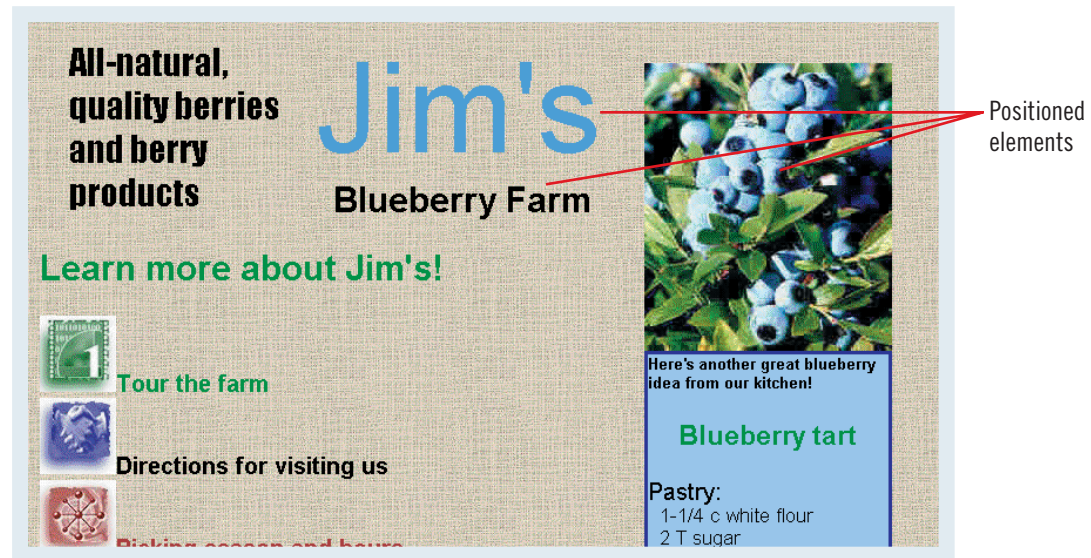
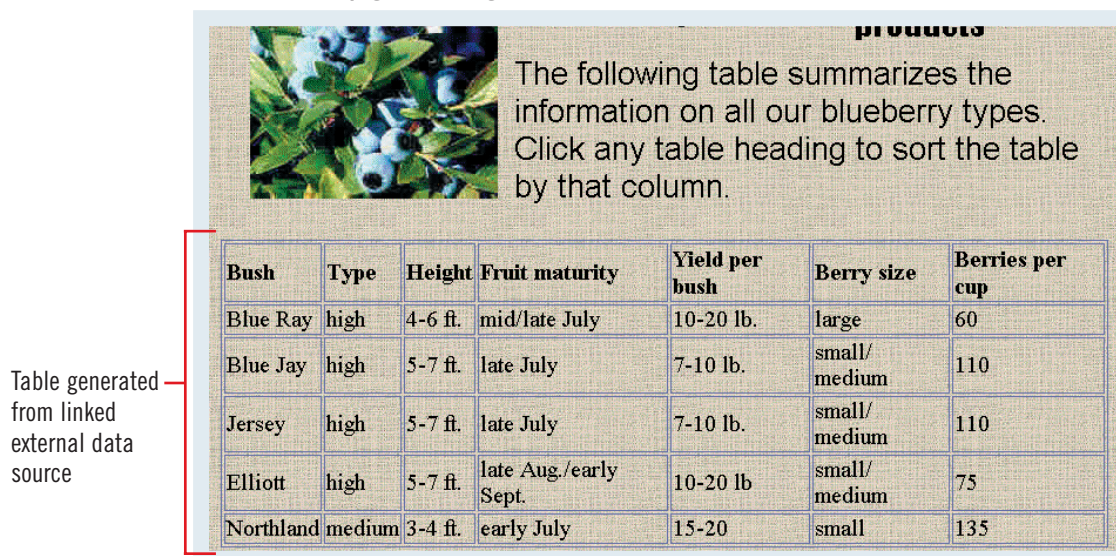


FIGURE I-5: Dynamic content and style changes



FIGURE I-6: Web page containing bound data





Understanding the DHTML Object Model

Developers of early scripting languages created an **object hierarchy**, which is a system of organization that allows Web page developers to describe and work with the Web page elements in a browser window. This hierarchy, officially called the **Document Object Model (DOM)**, categorizes and groups Web page elements into a tree-like structure. Each part of this structure is referred to as an **object**. For example, in the basic JavaScript DOM, a page's images, forms, anchors, and links are all grouped beneath the document object. The document, its location, and its history are, in turn, grouped below the frame object. DOMs allow browsers to identify page elements and to make them available to scripts in Web pages that they display. Although the earliest DOMs were part of scripting languages, Navigator 4 and Internet Explorer 4 have increased the range and versatility of DHTML by including their own extended DOM versions in the browser code itself, which are sometimes referred to as **DHTML Object Models**. DHTML Object Models allow you to reference a particular object the same way in any scripting language on a particular browser. However, because Netscape and Microsoft have developed different DOMs, you must reference some objects differently in Navigator 4 than in Internet Explorer 4. Figure I-7 shows the basic structure of the DOM for Internet Explorer 4, which makes virtually all browser window elements available to scripts. In order to take full advantage of DHTML's capabilities, Lydia reviews the top level of object classes in the Microsoft DHTML Object Model.

Details



Location

The location object contains the URL of the current page.



Frames

The frames object contains a separate Window object for each frame in the current browser window. When the window is not divided into frames, this object is empty and the entire document contents are part of the document object. The Microsoft DOM also contains a frames collection within the document object, to reference its `<IFRAME>` tag.



History

The history object allows access to the browser's list of previously visited URLs.



Navigator

The navigator object makes information about the browser available.



Event

The event object allows interaction with the event currently being processed by the browser, such as mouse movement or the press of a button.



Screen

The screen object makes information about the user's screen setup and display available.



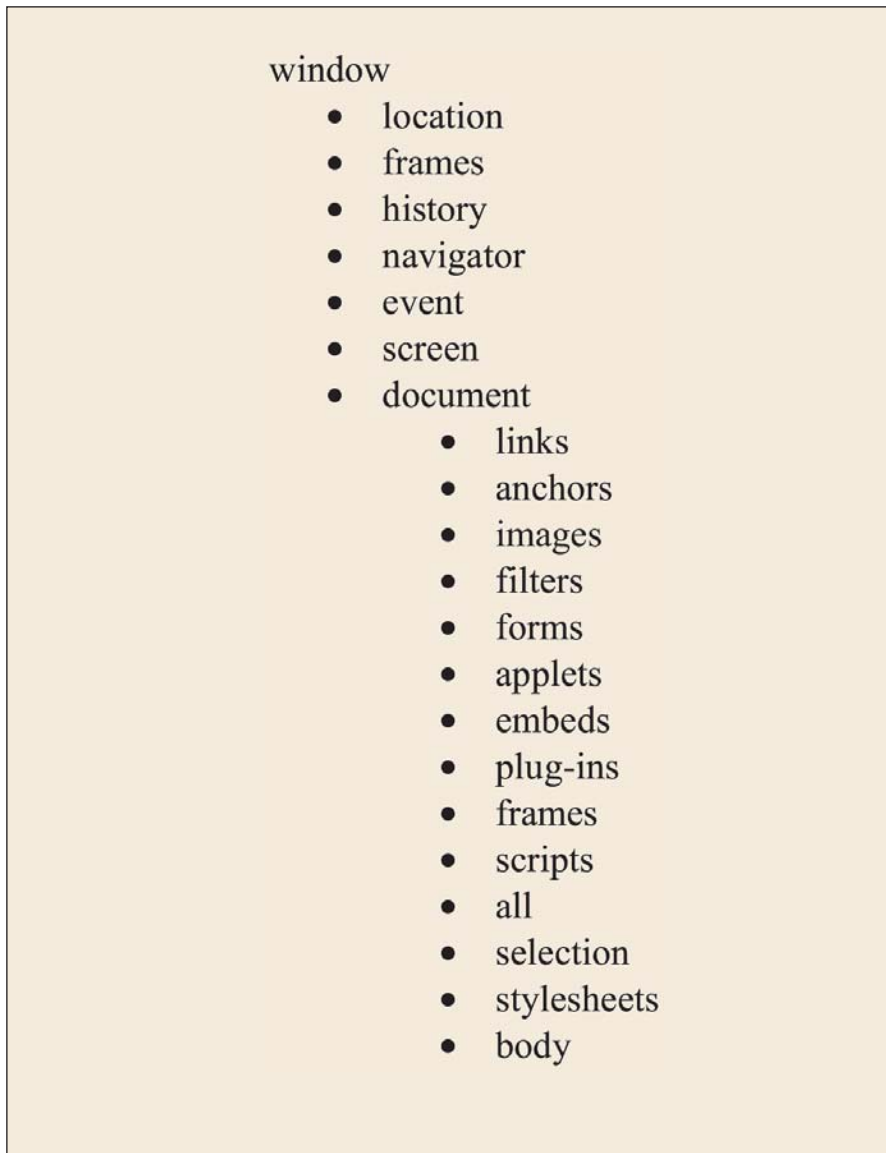
Document

The document object represents the current Web page in the browser window. A document object contains many elements as listed in Figure I-7. These elements, including links, anchors, images, and so on, are what help to give each Web page its unique characteristics.

Trouble?

Don't worry if you don't know the meaning of each element of the Document object. You will learn about some of these elements as you learn more about DHTML.

FIGURE I-7: Microsoft DHTML Object Model





Understanding Browser Variability

Although DHTML has few current standards, work is underway to change this situation. The **World Wide Web Consortium**, or the **W3C**—an international body whose mission is the creation of standards for WWW technologies—is creating official guidelines for DHTML. Although the W3C has created standards for the DOM and other DHTML technologies, these agreements have not yet resulted in a uniform interpretation of DHTML on the Web. The Microsoft and Netscape corporations, as the manufacturers of the vast majority of browsers in use today, have the greatest influence in DHTML implementation on the Web because the technology depends on each user's browser to interpret and run it. This difference in implementation means that code written for use on one browser may not work on the other, which requires writing two sets of code to incorporate some features into a page. As standards evolve and new browser versions are released, it will become easier to create today's features with a uniform code. However, both companies undoubtedly will continue to incorporate new, incompatible innovations into their browsers, which means that browser variability probably will always be a factor in creating DHTML pages. Lydia researches the implications of the different browsers available on the DHTML pages she is planning.

Details



Some dynamic HTML code is compatible with the 3.x versions of Navigator and Internet Explorer, known as **third generation** browsers, but most features work only on the 4.x and later versions, the **fourth generation** browsers. Because many users have upgraded to fourth generation browsers already and many more will upgrade eventually, it is often easiest to create DHTML with these browsers in mind. Remember, however, that if you want your pages to reach the largest possible audience, they must still accommodate other browsers. By organizing your pages to display logically even without their DHTML features and adding a few extra tags to allow older browsers to process the code, your content can remain accessible by older browsers that can't interpret DHTML, by text-based browsers, and by Web interfaces for people with disabilities. Testing your pages on different browsers before publishing them is important because standard DHTML code could cause older browsers to stop functioning, or **hang**. Figure I-8 shows a DHTML page in a fourth-generation browser and in Lynx, a text-only browser. Notice that the text-only browser ignores all of the DHTML commands, such as those for positioning, while still displaying all the information logically.



The differences in DHTML capabilities between fourth-generation browsers and earlier versions make writing interactive Web pages complicated because Internet Explorer 4 and Navigator 4 use and interpret DHTML differently. As you saw in the DHTML tour, therefore, they are not compatible when it comes to creating certain dynamic HTML features. Because DHTML components are still new technologies, many are not yet standardized in the software industry. For now, this incompatibility issue results in some features being available only in one browser and others being available only in the other. For some features to be available in either Web browser, DHTML Web page designers must write separate scripts to create similar features in the fourth-generation browsers.



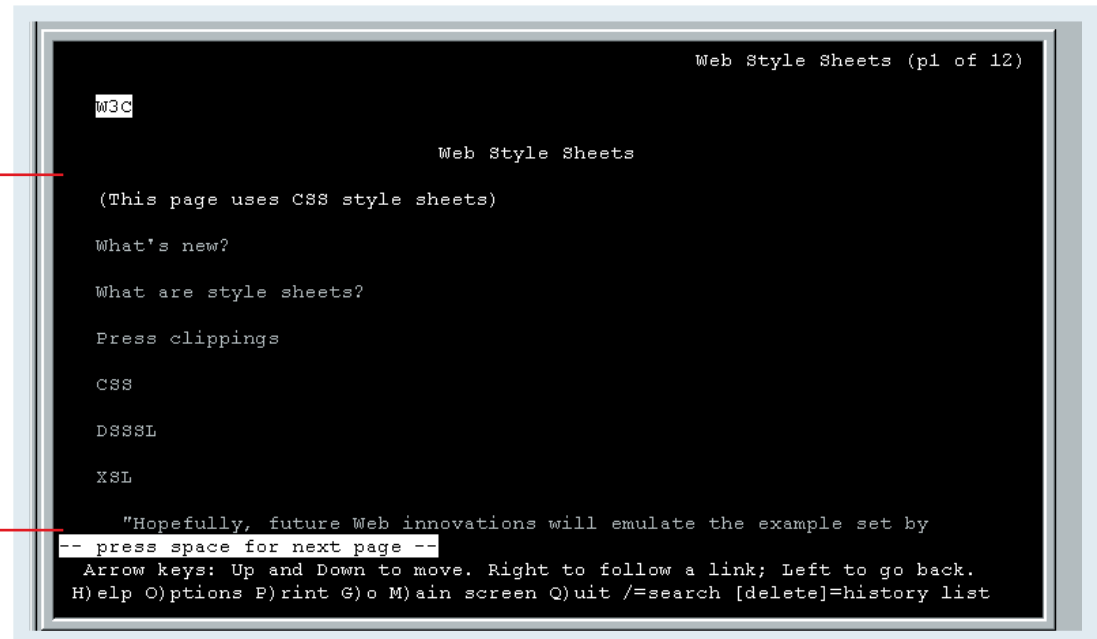
Designing dynamic Web pages is easiest when they will reside on a network where all users run the same browser, such as a corporate intranet. When publishing to the WWW, the only way to make sure that most users can view your pages is to write **cross-platform code**, or DHTML code that works on both fourth generation browsers. This often requires two sets of code in your page, along with a script to recognize in which browser the Web page is opening. Using this technique allows you to make interesting and interactive pages without causing compatibility problems for potential users. Although cross-platform coding can be time-consuming, many Web sites freely distribute such code that they have developed for popular features, along with tutorials describing how the code works. Using existing code can save a lot of page-development time.

FIGURE I-8: One Web page as it displays in two different browsers

Web page using style sheets and positioning in a fourth generation browser




Page structure and coding creates orderly display without DHTML features





Designing DHTML Pages

Like static Web pages, those incorporating DHTML require planning and forethought. Standard HTML rules, such as careful proofreading and judicious use of headings, still apply to dynamic pages. However, DHTML has its own advantages and pitfalls, which are important to keep in mind. In addition to awareness of browser differences, several other guidelines are helpful in working with this new technology.  To ensure that her Web pages follow good design principles, Lydia has made a list of recommendations, based on her DHTML research, for designing pages with DHTML.

Details



Organize for dynamic content

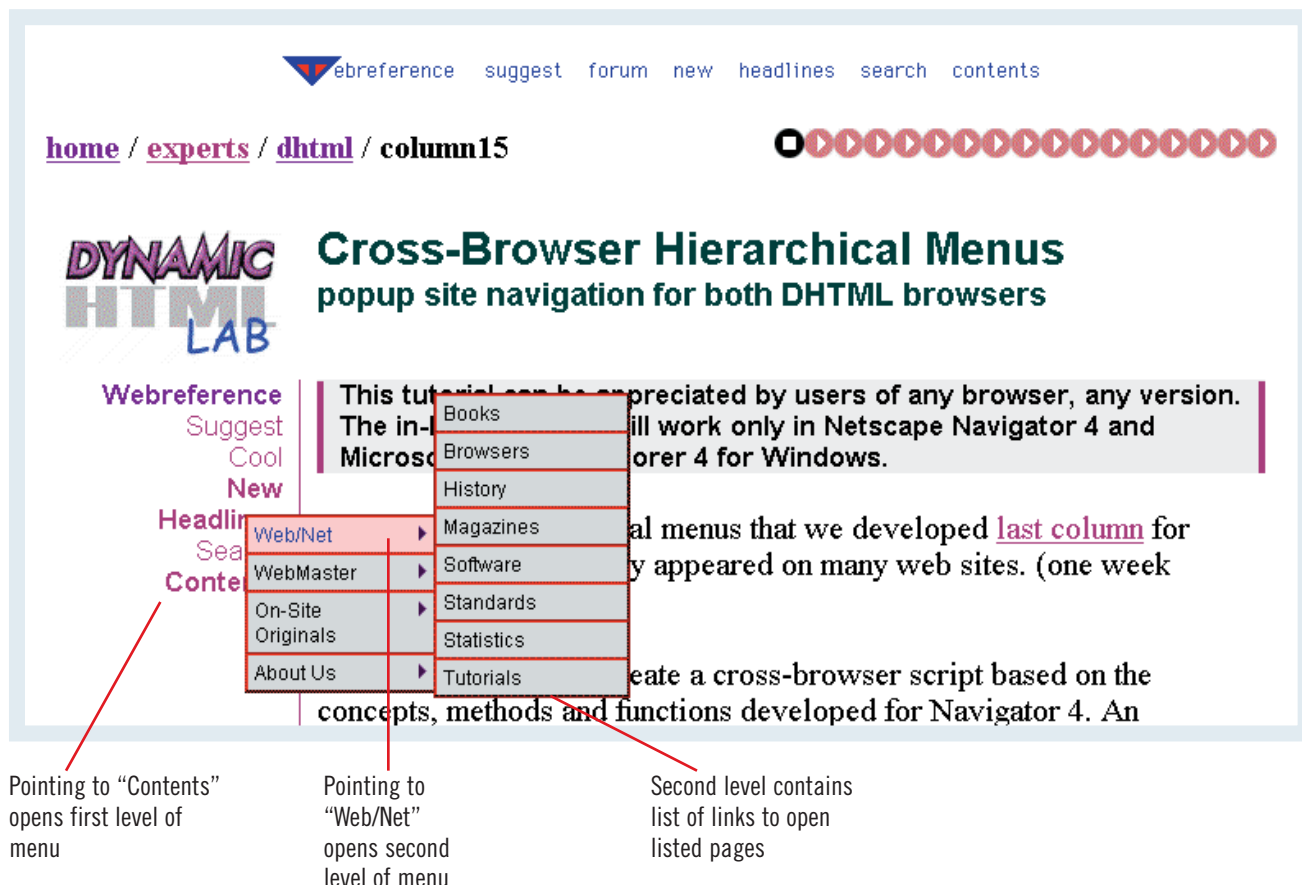
Remember that DHTML allows positioning of page elements and lets you show new content in response to user actions. Generally, this means that you can fit more in a dynamic page than in a static HTML page. For example, the hierarchical menu, shown in Figure I-9, allows a single page to contain information that otherwise works best as a list of links and a set of associated pages. This menu works like the one you use when you click the Start button in Windows. With DHTML, you can insert this within your Web pages, for simplified navigation, which keeps the Web page uncluttered. It also organizes the information so that the user can see the interrelationships of choices. Organizing your Web site to take advantage of DHTML capabilities can make the site easier for users to navigate and for you to manage.



Use dynamic features purposefully

Dynamic HTML features appear impressive, and you may be eager to show off your new skills by incorporating many of them into the pages you create. However, just as in static pages, the best Web pages are focused and free of distracting elements. Your Web page's content and message, rather than newly available features, should dictate which dynamic tools you use.

FIGURE I-9: Web page containing a hierarchical menu




Design resources

One advantage of the browser competition between Microsoft and Netscape is that both companies are eager to show off their browser's features. Both companies keep a list of links to well-designed pages

(supporting their own proprietary features) on their corporate Web sites. Reviewing these pages can give you ideas for planning successful dynamic Web pages as well as introduce you to new features.



Researching Code Architecture

After you outline your Web page and identify the DHTML features you want to use in it, the next step is to sit down and write code to make these features work. In many cases, although you have seen a Web page incorporating a certain action, it can be difficult to determine exactly how to create the feature with scripts and style sheets. At this point, research on the Web is indispensable to creating a successful DHTML page. For example, you can use the Web to look at the page source of a Web page that uses a feature you like. You also can find well-documented sample code on Web sites, which often you are allowed to modify and use for your own purposes.  Lydia has written the HTML for a new Web page that lists questions and answers. She wants to use DHTML to create a collapsible list on her Web page. In a collapsible list, explanatory text appears only when a user clicks its associated heading. Recently, Lydia saw the collapsible list feature on a page while browsing the Web, and she downloaded a copy of the Web page so that she could examine the code further.

Steps 1 2 3 4

1. Open the file **HTML I-3.htm** in your Web browser
The page displays a list of blueberry breeds.
2. Click the phrase **Blue Jay**
Associated text appears below the berry name, as shown in Figure I-10. Notice in the address bar that clicking the phrase did not open a new page; rather, it simply triggered a change in the appearance of the current Web page.
3. Start your text editor program, then open the file HTML I-3.htm to display the source code for the Web page
Figure I-11 shows part of the script for creating the collapsible list.
4. Close the text editor without saving the document
5. Be sure your computer is connected to the Internet, then open a search engine of your choice
6. In the “Search” text box, type **DynamicHTML programming** and click the **Find button**, or its equivalent
The browser returns a list of links to sites related to DHTML programming.
7. Review the links returned by the search engine, then follow one to a site that seems likely to contain tutorials or sample code
Articles and sample scripts for dynamic HTML applications may be helpful in creating your own pages.
8. Scan the site’s opening page for links to script libraries or articles about DHTML features, then follow the appropriate links
9. Locate sample code or a relevant article for an interesting DHTML feature, then download the page to your Project Disk with the name **Feature download.htm**
This downloaded file can be a helpful reference when you plan your own DHTML applications.

FIGURE I-10: Web page containing collapsible list

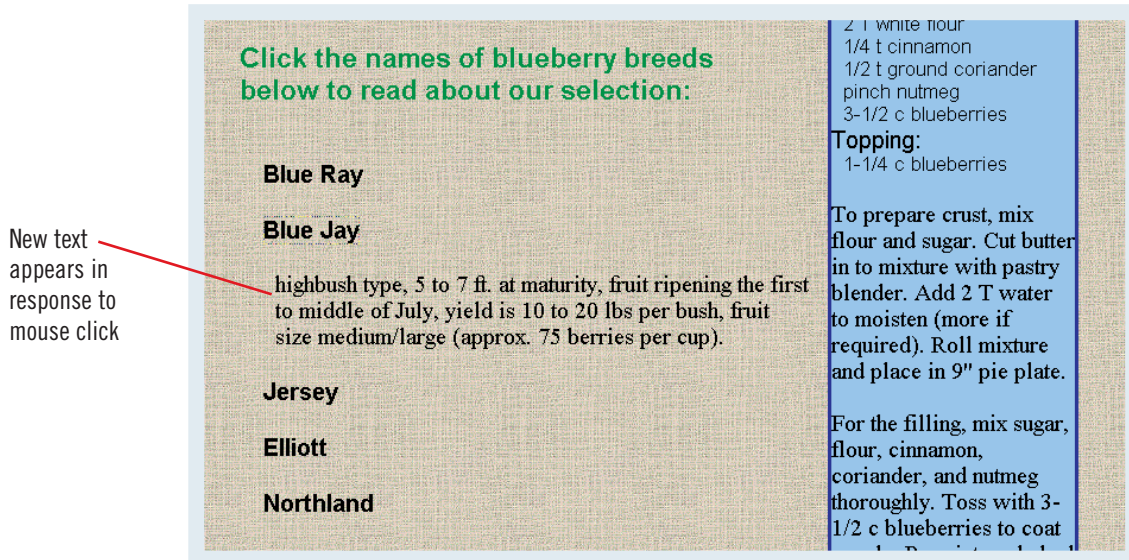
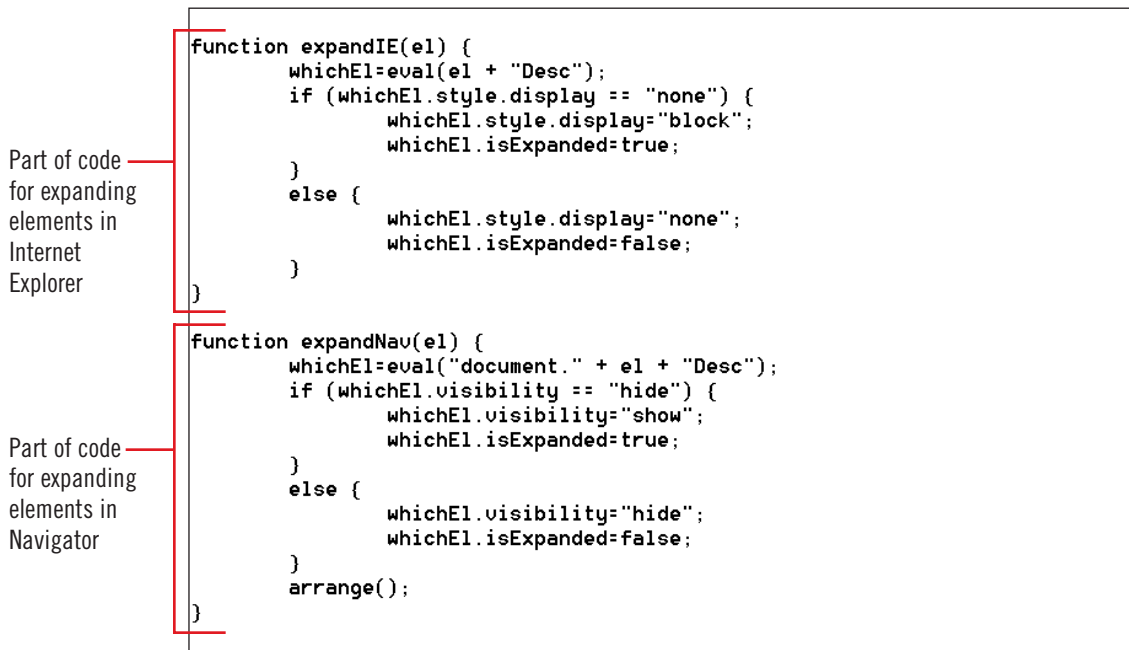


FIGURE I-11: Source code for expanding table of contents



Code-borrowing etiquette

When you find existing code on the Web that fits a project you are working on, it can save you time and frustration to use the existing code in your page instead of creating it from scratch. However, to be considerate to other designers, you should follow a few simple guidelines. First, be aware that some DHTML code is copyrighted and you cannot use it without permission from its author, which usually involves paying a fee as well. Some pages and sites offer code for free re-use. In this situation, it is still

considered courteous to credit the source of the code in your Web page, usually with the creator's name and the source URL. If you find code you'd like to use and are unsure whether you are allowed to, it is best to contact the creator for permission. If you don't have permission to use someone else's code, you can still use its basic framework to help you plan the creation of your page and then augment the features with your own coding.



Keeping Up with DHTML Changes

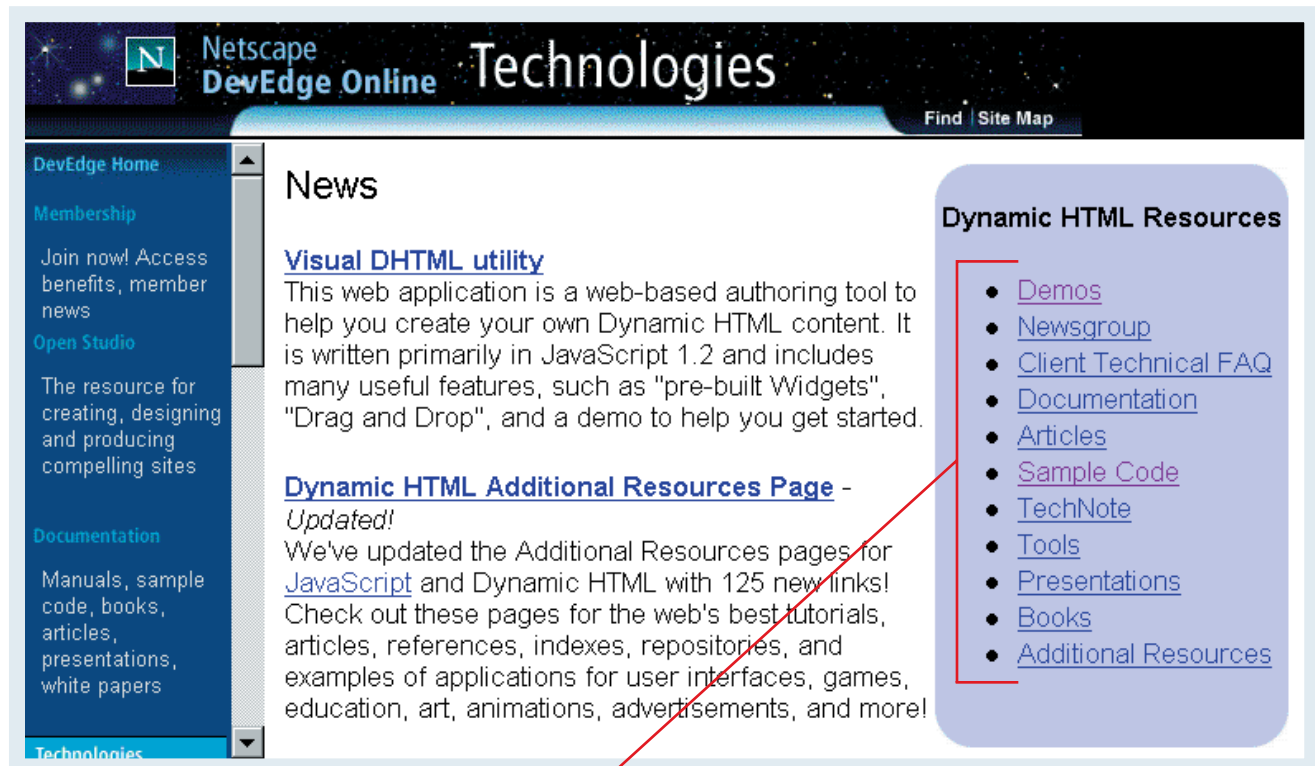
Web and software designers have already developed many ideas and methods for using DHTML in Web pages. As with any new technology, this body of knowledge will continue to grow, and in the process, will provide new uses and workarounds for DHTML programming. In addition, browser creators update and expand the capabilities of their products, resulting in ever-expanding possibilities for new DHTML applications. As a consequence of all these factors, it's important to stay current with the latest new developments in DHTML if you want to take full advantage of its possibilities for your Web pages. Predictably, the Web is a rich source of information on DHTML.

Steps 1 2 3 4

 Lydia wants to see what new DHTML features are on the horizon.

1. Be sure you are connected to the Internet, open your browser, then use a search engine of your choice to search the Web for the keywords **DHTML news**. The search engine returns descriptions and links to pages about DHTML.
2. Follow a link on the search results page to a site containing DHTML information
3. Scroll through and scan the opening page for tips on working with DHTML and for news about recent and upcoming developments
Figure I-12 shows a Web page offering tips and articles on using DHTML.
4. Follow links to explanations of new DHTML features or to news about upcoming additions or changes, then read one of these articles
5. Close your Web browser

FIGURE I-12: Web page containing DHTML tips



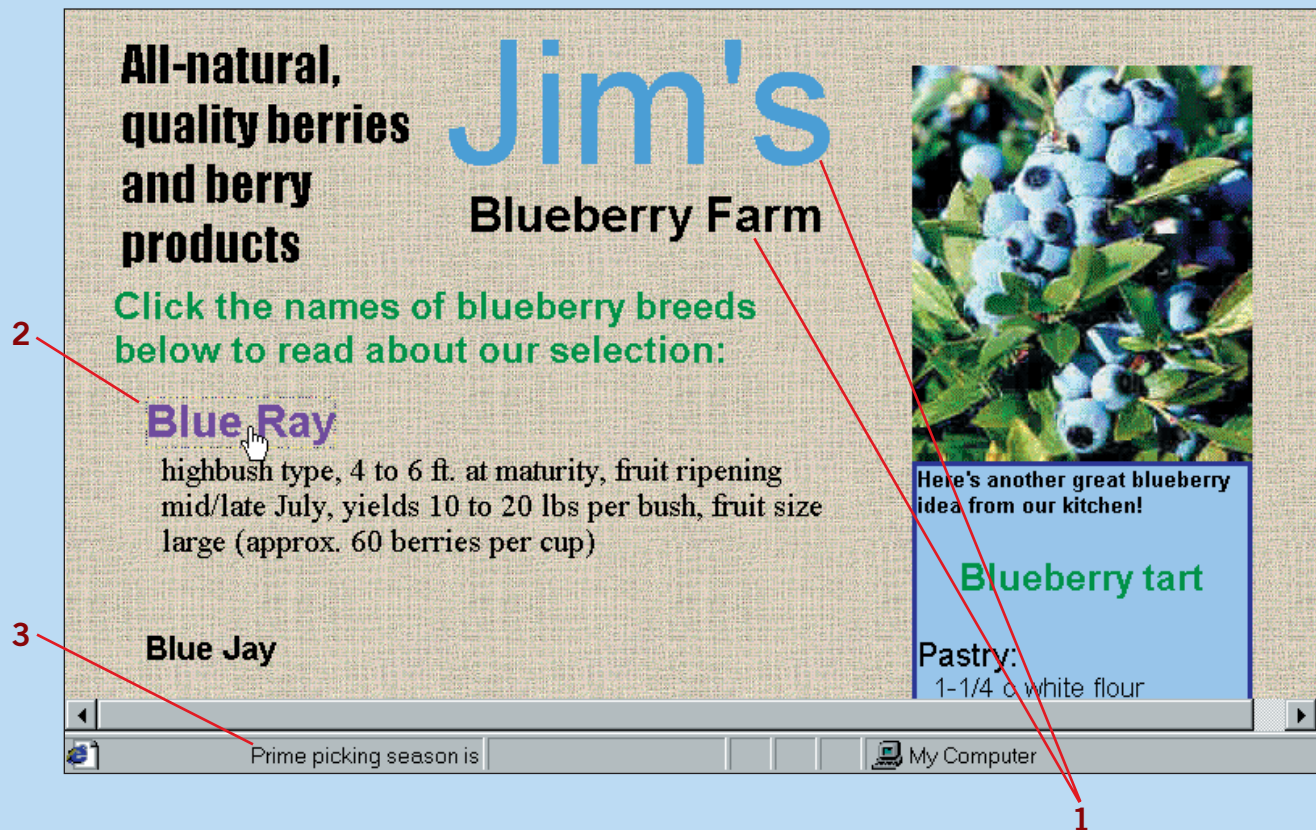
Resources available
on this site

Practice

► Concepts Review

Label each item in Figure I-13 with the DHTML category that best describes it.

FIGURE I-13



Match each term with its description.

- | | |
|---------------------------------|--|
| 4. Dynamic style | a. Hierarchy organizing browser window elements |
| 5. Static HTML | b. Collection of Web page technologies allowing quick response to user actions |
| 6. Client-side scripts | c. Scripts that the browser itself interprets and runs |
| 7. Cascading Style Sheets (CSS) | d. International body creating Web standards |
| 8. DHTML Object Model | e. Web page technologies allowing very limited interactivity |
| 9. Cross-platform code | f. Ability to specify locations of all Web page elements |
| 10. W3C | g. Component of DHTML allowing precise Web page style specification |
| 11. Positioning | h. Style that changes in response to user actions |
| 12. Dynamic HTML | i. Code that works on both fourth generation browsers |

Select the best answer from the list of choices.

13. Which of the following is *not* a feature of DHTML?

- a. Dynamic style
- b. Dynamic content
- c. Server-side scripts
- d. Data awareness

14. Positioning allows Web page designers to

- a. Create interactive page formatting.
- b. Create predictable layouts.
- c. Download Web page data.
- d. Create interactive page content.

15. A collapsible list is a good example of

- a. Dynamic style.
- b. Dynamic content.
- c. Data awareness.
- d. Absolute positioning.

16. DHTML uses an object model called

- a. The World Wide Web Consortium.
- b. JavaScript.
- c. Cascading Style Sheets.
- d. The DHTML Object Model.

17. Fourth generation browsers include

- a. Internet Explorer 5.0.
- b. Navigator 4.0 and Internet Explorer 4.0.
- c. Navigator 3.0 and Internet Explorer 3.0.
- d. Lynx and other text-based browsers.

18. Creating DHTML pages for both fourth generation browsers requires

- a. Excluding CSS from your pages.
- b. Limiting the pages you write to working only on one browser.
- c. Eliminating all dynamic features from your pages.
- d. Using cross-platform code.

19. What should provide the underlying structure for your Web pages?

- a. The page's content and message
- b. The amount of information you want to include
- c. Other pages you see on the Web
- d. The dynamic features you want to include

► Independent Challenges

1. The owners of the Green House plant store have asked you to add to their Web site a list of houseplant products they sell, along with a description of each. You think that a collapsible list would format and display this information easily and concisely. To begin, you design this Web page on paper.

To complete this independent challenge:

- a. On a sheet of paper, write the text for one or more titles for the Web page.
- b. Below the headings, copy the following list of products, along with description placeholder text:
 - Potting soil
 - [description]
 - Washed gravel
 - [description]
 - Peat moss
 - [description]
 - Houseplant fertilizer
 - [description]
 - Cactus soil mix
 - [description]
- c. On your Web page outline, label each element to indicate what page elements will be part of the collapsible list and what page elements will be appear on the Web page at all times.
- d. Indicate how you will format each text item.
- e. Indicate next to each of the product names that it will be formatted with dynamic style and add a second style specification for how the text will display when a user interacts with it.
- f. Indicate on your sketch any positioning you will use in your page.
- g. Add any further text to your sketch, such as that for hypertext links, and label the text with its formatting specifications and any additional features.



2. Sandhills Regional Public Transit wants to discuss with you ways to make their Web pages more interactive. In preparation for meeting with your clients, you want to become more familiar with different DHTML formatting options.

To complete this independent challenge:

- a. Log on to the Internet and open a search engine of your choice, then search on the phrase *DHTML formatting*.
- b. Click one of the links provided by the search engine to open a Web site containing DHTML formatting resources.
- c. Scan the opening page and navigate the site to locate articles or sample scripts for Web page features using DHTML.
- d. Read and print the article or the explanatory text accompanying a script.
- e. Write a paper detailing two DHTML formatting features you think would be useful in a home page that provides information to a wide range of people. Explain why you would include these features and how they would enhance the page.



3. Community Public School Volunteers has hired you to manage their Web site on an ongoing basis. To stay on top of the latest Web page design trends, you want to regularly research relevant news on the Web. Because you're preparing to create dynamic pages for CPSV, you want to research the state of W3C standards for DHTML.

To complete this independent challenge:

- a. Log on to the Internet and search on the keywords *W3C DHTML standards*.
- b. Investigate the sites listed on the search results page. Locate and print two articles regarding recommendations or standards released within the past six months.
- c. For each article, use the Microsoft and Netscape Web sites to research whether the standard is supported in each company's fourth-generation browser or if the company has announced plans to comply with the standard in a future release.
- d. Write a paragraph on each article, summarizing the area of DHTML it covers (for example, scripting or dynamic style), which browsers support it or will support it, and an overview of the article's content.



4. Explore sample DHTML pages on the Web, either at the Microsoft or Netscape Web sites, or by searching on the term *DHTML sample pages* in a search engine. Choose one Web page, print it out from your Web browser, and on the printout areas of the page that demonstrate DHTML features. On a separate sheet of paper, list these elements and, if applicable, describe briefly how they respond to user actions. Submit your printout and list to your instructor.